Modeling Phishing Decisions using Instance Based Learning and Natural Language Processing

Tianhao Xu
University of Washington
tx29@uw.edu

Kuldeep Singh
Carnegie Mellon University
kuldeep2@andrew.cmu.edu

Prashanth Rajivan
University of Washington
prajivan@uw.edu

Abstract

*Phishing is the practice of deceiving humans into disclosing sensitive information or inappropriately granting access to a secure system. Unfortunately, there is a severe lack of theoretical models to adequately explain and predict the cognitive dynamics underlying end-user susceptibility to phishing emails. This paper reports findings from an Instance-Based Learning (IBL) model developed to predict human response to emails obtained from a laboratory experiment. Particularly, this work investigates the effectiveness of using established natural language processing methods, such as LSA, GloVe, and BERT, to represent email text within IBL models. We found that using representations that consider contextual meanings assigned by humans could enable IBL agents to predict human response with high accuracy (*80%*). In addition, we found that traditional NLP methods that capture semantic meanings in natural language may not be effective at representing how people may encode and recall email messages. We discuss the implications of these findings.*

Introduction

Although phishing attacks are rampant on the internet, the likelihood of an individual encountering a phishing attack on a given day is small. Yet, people are expected to detect such a rare attack when they do experience one. Distinguishing phishing emails from legitimate emails remains a difficult task for a majority of people because phishing attacks are essentially *deceptive* messages that: a) are rare and constantly evolving; b) use impersonation to resemble truthful messages; c) applies emotional arguments to influence recipients; and d) could be tailored to exploit life context and recent world events [1, 2].

Despite the large body of research on phishing attacks, there is a lack of models that explain the key cognitive processes governing end-user response to phishing attacks. Existing research on phishing has predominantly focused on: developing solutions to automatically detect phishing emails [3]; testing whether people pay attention to essential cues in a phishing email or website [4]; developing interventions to aid human attention [5]; and developing training programs to educate people about the concepts and strategies related to phishing (e.g.,[6, 7]). Central to many of the past research is the aspect of human attention, or the lack thereof [8, 4]. An individual's lack of attention towards key indicators, such as the URL (Universal Resource Locator) of a website, and sender address in a phishing email, is widely considered why end-users fall prey to phishing attacks [4]. However, human attention is intimately linked to the contents of human memory [9, 10], and there is a severe lack of models that explain the role of such cognitive processes (e.g., memory activation dynamics) on end-user susceptibility to phishing emails. Our hypothesis is that people make decisions on phishing messages based on past experiences by activating pertinent memories of decisions made in response to similar emails in the past.

To test this hypothesis, we developed a cognitive model based on Instance-Based Learning Theory (IBLT) of experiential-based decisions [11]. The IBL cognitive model was developed in Python (PyIBL [12]). In the current research, we developed a model to understand how people may process phishing messages in memory and to determine the influence of past experience on end-user decision making. The objective here is to understand the cognitive processes driving end-user response to phishing attacks. Such cognitive models could be potentially used within email applications (e.g., Microsoft Outlook) to predict how people may respond to novel phishing samples which could inform embedded phishing training and phishing risk assessments. These models, however, may not be useful for discriminating phishing from legitimate emails.

In the following sections, we first summarize the laboratory experiment procedure used to collect human responses to legitimate and phishing emails. Then, we

URI: https://hdl.handle.net/10125/79609
978-0-9981331-5-7

describe the methods used for developing a cognitive model to predict human response to phishing emails in the laboratory study. Finally, we present the results and discuss implications.

Email Management Task

To measure end-user susceptibility to spear phishing messages, we conducted a group experiment in the lab [13]. Each group in this experiment consisted of four participants. Among the four players, three players were randomly assigned the role of an end-user and the remaining participant in the group was assigned the role of an attacker.

End-user participants in the experiment were asked to assume the role of a fictional character during the experiment. They were presented with a detailed fictional narrative describing that end-user (e.g., fictional name, occupation, location). The end-user participants were then asked to perform a routine email management task pretending to be a fictional character and make decisions on emails on behalf of the character. Emails included legitimate emails addressed to that fictional role, promotional emails, mass phishing emails, and spear-phishing emails targeted to that fictional role. We randomly selected the names of three people from the Enron dataset and 70 emails they received [14]. These email messages served the purpose of providing the necessary context for the end-user in the experiment. Additionally, promotional and mass phishing emails were also provided to end-user participants, randomly chosen from the data sets used in past studies [10, 15]. The study protocols were reviewed and approved by University of Washington's Institutional Review Board(IRB) office.

The end-users in the experiment were simply making decisions on whether they would respond or not to any given message presented to them. They were rewarded based on their performance in the task. For each email, the end-user participants also responded to a small survey (see Table 1) about the email content. The questions in the survey were initially developed and improved in a previous study [15].

The participant playing the role of an attacker in the group was given specific goals in the experiment using the available information to them about the target end-user. The attacker objective was to steal bank credentials, work account credentials, and lure the target to download attachments. These were fake objectives, and no real harm were caused to any participants. If the attacker was able to deceive end-user successfully in an email, i.e., end-user responded to that email than the attacker earned rewards. The overall goal for the adversary was to maximize their individual rewards in the experiment.

Survey	Summary
Request for action (Task assigned, click on a link, download attachment, etc)	Action
Request for information or opinion (send a reply message, contact info,send file, image, etc)	Information
Contains status update for an ongoing project or task	Project
Request for a meeting or other communication with you	Meeting
Contains reminder for a meeting, event, or upcoming deadline	Deadline
Spam or marketing or suspicious	Spam
Other	Other

Table 1: Survey questions presented to end-user during each trial

Dataset

To model phishing decisions, we leveraged the dataset generated from the experiment described in the previous section. This experiment was designed to understand the various factors involved in the spear-phishing attack. The dataset from this experiment contained responses from 84 participants. The dataset consists of a total of 529 unique emails with 6712 responses. Each participant responded to approximately 80 emails, including benign emails, phishing emails, and spear phishing emails. For full information about the study, please refer to this previous publication [13].

Cognitive Model

We developed an end-user cognitive model using an Instance-Based Learning (IBL) model of binary choices [16, 17] in python using PyIBL [12]. IBL models use the formalization of the memory mechanisms from the adaptive control of thought-rational (ACT-R) cognitive architecture [18] and the decision process from Instance-Based Learning Theory (IBLT) [11]. An instance in the IBL model is a unit of experience, consisting of the state (attributes of task), the decision made in the current state, and the utility (the outcome of choosing an option in the current state). For each viable decision, the model computes an expected utility using the *blending* mechanism. The blended value is computed by averaging the past outcomes weighted by the probability of memory retrieval, which depends on the contextual similarity to past instances. It also

take into account frequency and recency of the past experience instances. The decision is made with the highest expected utility. To calculate blended value $V_{k,t}$ for option k at trial t the following equation is used:

$$V_{k,t} = \sum_{i=1}^{n} P_{i,k,t} * X_{i,k,t} \quad (1)$$

Where $X_{i,k,t}$ represents the outcome of an instance i for option k at trial t and $P_{i,k,t}$ is the retrieval probability of an instance i for option k at trial t. The retrieval probability of an instance i is the ratio of activation of i_{th} instance corresponding to the activation of all instances (1, 2, ..., n; where n is total number of instances) created within the option k at trial t. The retrieval probability is defined as:

$$P_{i,k,t} = \frac{e^{A_{i,k,t}/\tau}}{\sum_{i=1}^{n} e^{A_{i,k,t}/\tau}} \quad (2)$$

Here, $\tau = \sigma * \sqrt{2}$ and τ is a free noise parameter. The noise parameter (τ) is used to capture the inaccuracy of remembering past experiences from memory. $A_{i,k,t}$ is the activation of an instance i on option k at trial t. It represents the linear aggregation of three cognitive elements: frequency and recency, similarity of an instance with past experiences, and noise. Based on the ACT-R theory of cognition [19], the activation value represents how readily available an instance is in memory: the higher the activation, the easier and faster it would be to retrieve such an instance from memory. The Activation is computed as follows:

$$A_{i,k,t} = ln \sum_{t_i=1..t-1} (t - t_i)^{-d} + MP \sum_{k} Sim(v_k, c_k) + \sigma * ln\left(\frac{1-\gamma_{i,k,t}}{\gamma_{i,k,t}}\right) \quad (3)$$

The term ($ln \sum_{t_i=1..t-1} (t-t_i)^{-d}$) reflects the power law of experience and forgetting, t_i represents all the previous trials where the instance i was either created or its activation was reinforced due to its recurrence. t_j is the time since the j^{th} occurrence of instance i and d is the decay (default value= 0.5) rate of each occurrence. The decay parameter accounts the the rate of forgetting the experienced events: higher the decay, faster the rate of forgetting of past events, which increase the reliance on recent events. The activation of an instance can increase with the frequency and recency of observing that outcome (i.e., by small differences in $t - t_i$).

$MP \sum_k Sim(v_k, c_k)$ represent a partial matching process which reflect the similarity between the current state (c_k) and the instances that are stored in memory(v_k), scaled by a mismatch penalty (set to 2.5). The similarity between numerical slot values are computed on a linear scale from distinct (0.0) to an exact match (1.0).

$\sigma * ln\left(\frac{1-\gamma_{i,k,t}}{\gamma_{i,k,t}}\right)$ represents the Gaussian noise mechanism for capturing the variability in individual choices. Where $\gamma_{i,k,t}$ is a random number drawn uniformly between 0 and 1. The σ i.e. the variance in the noise term is set to the default ACT-R value of 0.50.

In a related work, Cranford et al. developed an instance-based learning model to predict end-user response to phishing emails[9]. LSA (Latent Semantic Analysis) and Wordnet were used to compute the semantic similarity between incoming emails and agent memories. However, using LSA to determine semantic similarities between two instances was raised as an important limitation of this work. They theorized that LSA was not effective at representing how people process email texts. Therefore, in this work, we investigate the effect of using deep learning and attention-inspired natural language processing methods such as BERT, which has demonstrated impressive performance in other NLP applications.

Natural Language Processing Methods

We tested three different natural language processing methods (LSA[20], GloVe[21], and BERT[22]), often used in natural language understanding tasks, to represent and calculate the similarity between two email instances within the IBL model. We compared the performance of these three methods in predicting participants' responses to emails (ham, phishing, and spear-phishing) in the laboratory study described earlier. In the following section, we briefly describe each method and how these three methods were used to measure similarity between instances in the IBL model.

As a baseline, we used *Latent semantic analysis* (LSA). It is a popular bag-of-words approach used to determine the similarity between two linguistic items (e.g., documents, emails) based on word frequencies. In LSA, linguistic items are organized into a word-frequency or a TF-IDF (term frequency-inverse document frequency) matrix to specify the number of times each word in a corpus appears within each document. Using the singular value decomposition method, this large dimensional matrix is factorized to represent the documents in a low dimension space and determine the latent factors (or topics) that may describe each document. The similarity between a pair of documents is calculated using the cosine distance between the low-dimension vector of latent factor values of each document. LSA effectively captures the similarity between documents based on word frequency that may suit information retrieval applications. However, they may not represent how humans process text. For example, two emails from the

same bank could contain similar words (e.g., account, withdrawal) and branding but could be communicating two different things. A human would consider the two emails dissimilar to each other, but LSA is likely to consider them semantically similar because they contain words that belong to a common latent topic of banking.

GloVe or Global Vectors algorithm was introduced to address some of the limitations in LSA and other algorithms (e.g., Word2Vec) used for learning word-level representations [21]. GloVe algorithm is used to produce a global vector representation of words based on their co-occurrence in a large corpus of text data. The algorithm generates a vector representation for each word in a given corpus, where words with similar vector representations are considered semantically similar. For example, let us consider two words, chase and account related in the banking context. The global vector representations of these two words are essentially the ratio of their co-occurrence probabilities with various probe words. When fine-tuned to the banking context, the algorithm is likely to generate similar vector representations for the words chase and account due to their co-occurrence in a banking email corpus. The algorithm generates a vector of fixed dimensions (usually 300) for each word in a corpus. These are machine-learned representations, learned by training the algorithm on a large text corpus to capture global co-occurrence statistics between words. Using the transfer learning approach, the global representations can be fine-tuned and applied to learn representations for words in a smaller dataset, such as a phishing dataset generated from the experiment described earlier. Although GloVe is significantly better at capturing global and contextual similarities between words, the sequential pattern of language and the context of a word within a sentence is ignored.

We have used Bi-directional encoder representation (BERT) as a third approach to represent the email text within the IBL model [11]. Unlike LSA and GloVe, which represent natural language at a word level, BERT can be used to make inferences at a sentence level, allowing BERT and other related methods to achieve state-of-the-art performance on various natural language understanding tasks [22]. BERT is a bi-directional model because the algorithm considers the full context of a word in a sentence by processing words that come before and after it. The algorithm achieves bi-directional processing by considering all words in a sentence in parallel rather than one-by-one in a sequence, using a transformer-based self-attention mechanism originally introduced in the paper [23]. The architecture of BERT is complex to describe succinctly and is beyond the scope of this paper. However, in essence, the self-attention mechanism used in BERT takes inspiration from how humans pay visual attention to text stimuli such as words in a sentence - we fixate and associate relevant terms in a sentence and skip over irrelevant words. Similarly, the encoder module in BERT takes as input a long text sequence, encodes, and generates a word embedding for the input sequence, and using the self-attention mechanism, it recodes and weighs the relevance of words in the sequence to highlight the pertinent parts in the long text. For the full description of the BERT model, please refer to the original paper[23]. Like GloVe, we can use the transfer learning approach to fine-tune the BERT model for specific NLP tasks. Sentence-BERT is a modified version of the pertained BERT network that uses Siamese network structures to derive semantically meaningful sentence embeddings that enables us to compute the cosine similarity between a pair of email texts [24].

Partial Matching using NLP

The NLP methods described in the previous section were used in the IBL model to determine similarities between two email instances. Specifically, the NLP methods were used as sub-models for the similarity term $\sum_k Sim(v_k, c_k)$ that represent the partial matching process in the IBL model. The core idea is to use each of the three NLP methods to represent emails as fixed length numerical vectors. LSA and GloVe were used to produce vectors representing word-level embeddings, whereas BERT was used to produce feature vectors representing sentence-level embeddings. The similarity term in IBL is essentially defined as a cosine distance function measuring distance between the feature vector representations of the current email instance (c_k) and email instances stored in memory (v_k).

LSA In the LSA method, all 529 email messages obtained from the study were mapped into a matrix with 529 columns and 6213 rows. The number of rows represents 6213 unique words present in the corpus. A truncated singular value decomposition method was applied to reduce the number of dimensions to 300. The number of dimensions was chosen to be 300 to match the vector sizes derived from GloVe and BERT. The final LSA matrix was of size $529{*}300$. Each row in the matrix represents the feature vectors for each message derived using the LSA approach. Similarity is calculated using a cosine distance between two email vectors from the LSA matrix, using the equation below.

$$similarity = \frac{v_k * c_k}{\|v_k\|\|c_k\|}$$

GloVe Using transfer learning, we fine-tuned and applied the GloVe matrix to represent each of the unique words across the GloVe's pre-trained 300-dimension word vector. This produced a matrix of size 6213*300, where 6213 is the number of unique words in the corpus. We created another matrix ($529 * 6213$) that contained the number of times each of the 6213 unique words occurred in each of the 529 messages. The two matrices were then multiplied to produce a final matrix (529*300) that contained the 300-dimensions word vector representation for each message derived using the GloVe method. Similarities between two emails were calculated using a cosine distance function between the corresponding two feature vectors in the GloVe matrix.

Transformers Transformers has been used to achieve high performance in many natural language processing tasks. We used nli-distilroberta-base-v2 model, a distil RoBERT model, fine-tuned on SNLI(Standard Natural Language Inference) corpus[25] with 84.38 performance on the STS benchmark dataset. The SNLI is a large collection of human written English sentence pairs labeled for semantic similarity. This Sentence-BERT(SBERT) [24] was applied to our dataset to derive the semantic similarity between every pair of emails in the dataset. Figure 1 shows the architecture of the SBERT. For each email pair (c and v), the SBERT was applied using two identical BERT/RoBERTa models to process each email separately. Use the pooling layer, we derive a fixed sized length embedding vector (a and b) representing the respective emails. Like other methods, the cosine similarity function was used to calculate the similarity between the two vector representations of emails.

User Perception The three NLP approaches (LSA, GloVe, BERT) described earlier enable us to capture semantic similarities between two email instances based on underlying statistical properties. However, these methods may not be representative of how humans actually process text, recall text from memory, and make decisions in the email management context. During the experiment, we had asked participants to self-report their opinion about each message presented to them along seven dimensions that described what was in the message. For example, whether the message requests an action or whether the message contains a reminder for a meeting. Therefore, as an alternative approach, we represented each email along these 7 self-reported dimensions. The seven dimensions are listed in table 1. Emails with similar contents will have similar vector representations. These vectors represent how each individual participant perceived the contents of the message presented to them. Like with other methods, the similarities between two emails were calculated using a cosine distance function between the corresponding two self-reported feature vectors.

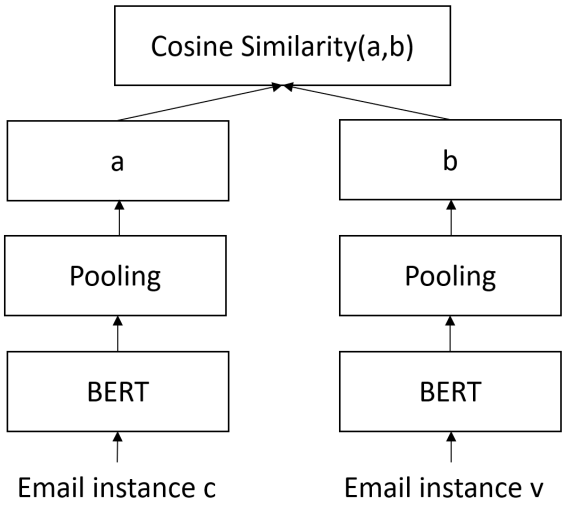

Figure 1: Sentence BERT architecture used to represent emails in IBL

Perception Bert Practically, it is not possible to train cognitive agents based on user-reported email attributes. Therefore, we re-trained the SBERT model to learn similarities between emails based on user perceptions. For each pair of emails, we fine-tuned the SBERT model to predict the similarity score measured using the user-reported attributes. The similarity score was used as classification and it ranged from 0 to 1. 0 indicates emails that are least similar according to users, and 1 indicates identical emails.

There were 529 unique messages in the dataset. Therefore, there were 529*528/2= 139,656 similarity pairs. We randomly sampled 10000 pairs of similarities to fine-tune the downstream task in BERT. 10000 pairs of similarities were split and trained using the ratio 7:1:2. We used a package from [24] to manage the computation involving BERT, and all computation was conducted on a desktop with RTX 2060 graphics processor unit.

Simulation Procedure

Dataset from a previous laboratory study was used to train the IBL agent and evaluate its performance in predicting human responses to phishing emails. The dataset includes 529 unique email messages and contains email responses from 84 participants. Among the 529 emails, 210 were ham emails, 15 were mass phishing emails, 52 were promotion emails, and 252 were spear-phishing emails. To simplify the response

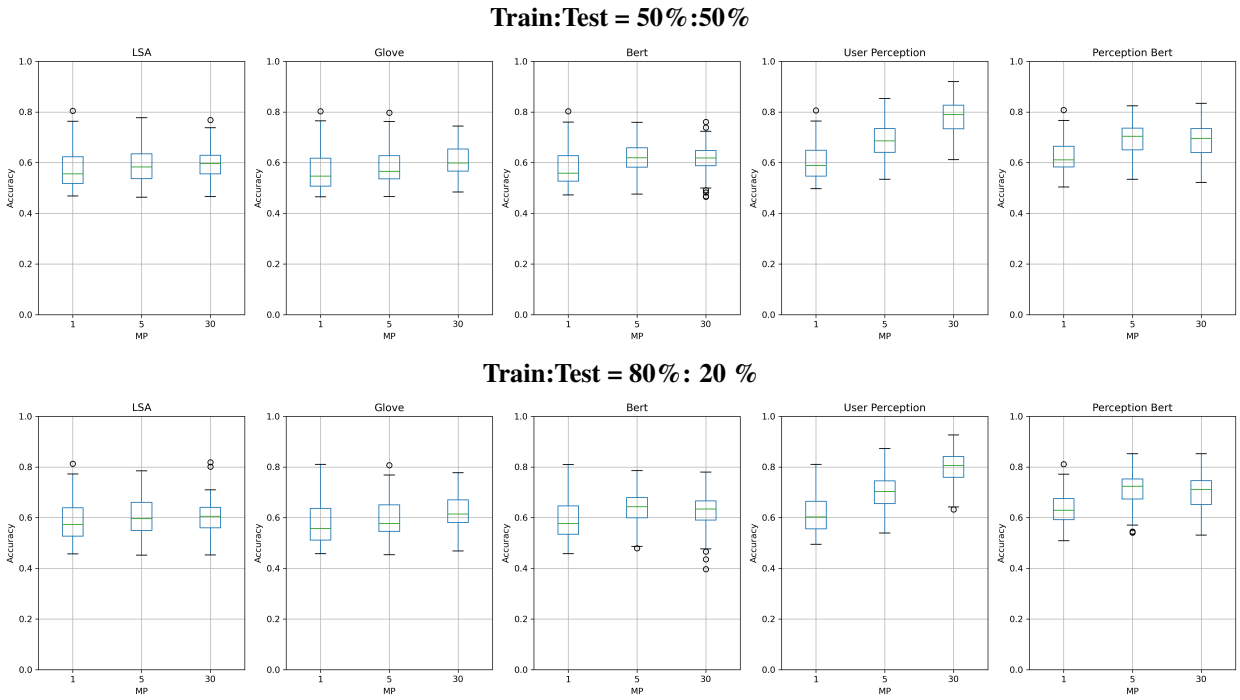

Figure 2: Accuracy for IBL agent across similarity approaches, MP values and split ratios

from end-users, we encode the response (a) Respond Immediately Or (b) Flag the email for follow-up to ***response*** and (c) Leave the email in the inbox, (d) Delete the email, Or (e) Delete the email and block the sender to ***ignore***. We trained IBL agents to model and predict responses from each of the 84 human participants in the laboratory study. Each IBL agent represented a single human participant and was presented with the same emails experienced by its human counterpart. On average, participants in the experiment made decisions on 80 emails which included ham emails, mass phishing emails, and spear phishing emails. Similarly, the IBL agents representing each participant made decisions on the same number of emails experienced by its human counterpart. For each email presented, the model takes as input the context of the email and generates an action (respond or not respond) by retrieving similar past instances. Typically, instances are encoded as chunks in an agent's declarative memory that represent the features of the decision: the context in which a decision is made, the action taken, and the outcome of that decision. In this work, the context was represented as a feature vector derived for each email message using the five similarity methods described in the previous section. This feature vector was provided as the input to the model. For each email feature vector presented, the model made a decision whether to respond or not. Each decision received outcome feedback: 1 point for correct response and -1 point for incorrect response. Except for the mismatch penalty parameter (see Equation 3), all other model parameters were set to their default values. For example, the decay parameter was set to the default value of 0.5. We experimented with three MP parameter values (1, 5, 30).

We also experimented with two split-ratios of training and testing: 50-50 and 80-20. With 50-50, the IBL agent was trained on 50% of randomly selected emails that its human counterpart experienced, for example, if a human participant made decisions on 100 emails during the study, the IBL agent representing the participant was trained on 50 randomly selected emails that the participant experienced. These emails served as the training instances for the agent and were encoded as instances in the declarative memory of the agent. The agent's decision performance was evaluated on the remaining 50% of the emails unseen by the agent during training. In the 80-20 split, the IBL agents were trained on 80% of randomly chosen emails the participant experienced and tested on the remaining 20% of emails. Since IBL is a stochastic model, the model was trained and evaluated 800 times to generate stable predictions of human behavior. Using the half-width approach [26], we estimated 800 model replications were necessary to achieve 95% confidence with our output estimates.

Results

We analyzed the performance of IBL agents in predicting human response to emails. We used model accuracy, hit rate, and correct rejection rate to measure model performance during the test phase. For each IBL agent representing a human participant, *accuracy* measured the percentage of emails for which the model decisions concurred with the human decisions. For example, a 70% accuracy would indicate that IBL agent accurately predicted the human response on seventy percent of emails presented during the test phase. In addition to the accuracy, the hit rate and correct rejection rates were also calculated. As shown in Table 2, for each IBL agent, the hit rate measured the proportion of emails for which the IBL agent accurately chose to respond to it, whereas correct rejection measured the proportion of emails for which the IBL agent accurately chose to ignore (not respond) the email.

		Human	
		Response	*Ignore*
IBL	*Response*	Hit	False Alarm
	Ignore	Miss	Correct Rejection

Table 2: Hit rate and Correct Rejection Rate

Model accuracy, hit rate, and correct rejection rates were calculated by averaging the performance of 84 agents (representing 84 human participants) across 800 model runs. Figure 2 compares the accuracy of IBL agents in predicting human response across the five similarity methods (LSA, GloVe, BERT, user perception, perception bert), three mismatch penalty parameter values (1, 5, 30) and the two split-ratios used for training and testing the agents (50-50 vs 80-20). Figure 3 shows the distribution of hit rate and correct rejection rate. Each point in the graph represents the average hit rate and correct rejection rate of one IBL agent predicting the response of its human counterpart. The rates are color-coded to indicate the five different kinds of similarity approaches compared. The average performance with three mismatch penalty parameter values and two split ratios were also compared and presented in the Figure 3.

Using mixed-effects ANOVA, we tested the effect of different similarity approaches, mismatch penalty value, and split-ratio on all three performance measures. Tables 3, 4, and 5 presents the results from the ANOVA analysis for the three performance measures: accuracy, hit rate, and correct rejection rate, respectively. Similarity approach and mismatch penalty value were found to have a significant effect on all three measures, whereas the split-ratio had a significant effect on accuracy and correct rejection rates. We will discuss each of these results in more detail next.

	Df	**F value**	**Pr(>F)**
MP	2	184.78	0.0000
Approach	4	228.73	0.0000
Split Ratio	1	20.45	0.0000
Approach:Split Ratio	4	0.06	0.9934
MP:Split Ratio	2	0.23	0.7938
MP:Approach	8	39.66	0.0000

Table 3: Anova table for Accuracy

	Df	**F value**	**Pr(>F)**
MP	2	194.29	<0.001*
Approach	4	122.72	<0.001*
Split Ratio	1	4.48	0.0344*
Approach:Split Ratio	4	0.15	0.9627
MP:Split Ratio	2	5.29	0.0051*
MP:Approach	8	11.55	<0.001*

Table 4: Anova table for Correct Rejection Rate

	Df	**F value**	**Pr(>F)**
MP	2	235.46	<0.001*
Approach	4	84.17	<0.001*
Split Ratio	1	0.00	0.9591
Approach:Split Ratio	4	0.12	0.9760
MP:Split Ratio	2	0.95	0.3861
MP:Approach	8	7.17	<0.001*

Table 5: Anova table for Hit Rate

Similarity Approach

Between the five similarity approaches we tested, we found that the approach that used participants' self-reports to represent emails within IBL model performed better than all other approaches (79.7% average accuracy). Following it closely, the second best performing model was the approach that used the perception Bert model, fine-tuned using participants self-reports. Although there is a significant difference in accuracy between the two approaches, we found that the perception Bert model performed, on average, only 2.23% ($p < 0.001$) lower than the model that directly used participants' self-reports to represent the emails. LSA, GloVe and canonical BERT, on average, achieved less than 60% accuracy in predicting human participant response. There was no significant difference in model accuracy between using LSA, GloVe, and BERTs. From Figure 3, it can be observed that the approach using participants' self-report to represent emails clustered at the top right corner indicating higher hit rates and higher

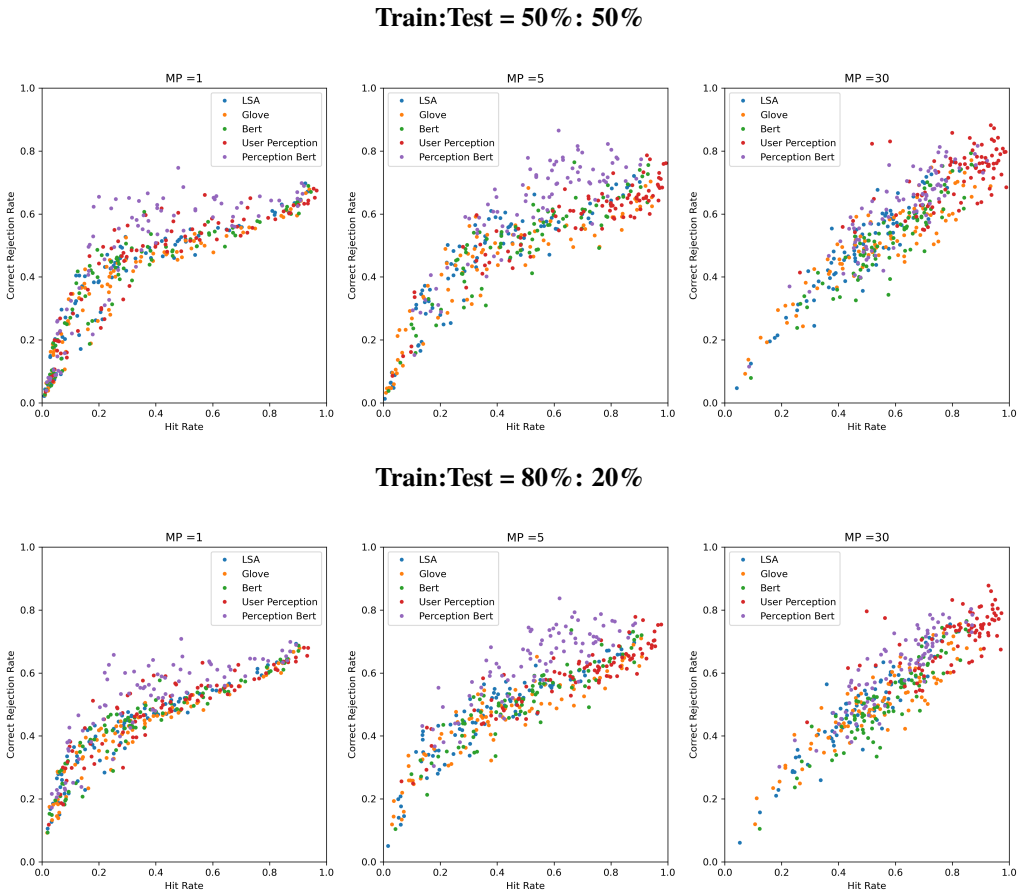

Figure 3: Correct Rejection Rate vs Hit Rate for IBL agent across similarity approaches, MP values and split ratios

correct rejection rates. The pattern is the same across all values of MP and split-ratio.

To further analyze the differences in performance, we compared the similarity scores (calculated using cosine distance function) between all pairs of emails derived using the GloVe approach against the similarity scores derived using participants' self-report. We compared these two approaches because the GloVe approach represents similarities calculated based on the statistical properties of text. In contrast, the user perception approach represents similarities calculated based on individuals' opinions about the email attributes. As shown in Figure 4, the GloVe approach considered the majority of the emails as similar to each other, whereas the user perception approach considered the majority of the emails to be dissimilar to each other. We calculated the correlation between the two matrices containing the similarity scores derived from the two approaches. The correlation between the two approaches was low, $r_{person} = 0.10, p < 0.001$. This indicates that for the same pair of emails, the similarity scores obtained using the two methods were different from each other. It is noteworthy that both approaches used the same cosine distance function to measure the similarity for any given vector pair. This shows that pure semantic methods such as LSA, GloVe and canonical BERT were ineffective in capturing the deep contextual differences present in emails, which may have been considered as a significant difference from a human perspective.

Mismatch Penalty

A large mismatch penalty(MP) value makes the IBL agent care more about the difference between the current instance and previous instances. Large MP values penalize agents more for incorrect mismatches between the incoming instance and the instance in memory. Overall, we found that the mismatch penalty has a significant effect on all three measures. A post-hoc analysis revealed that irrespective of the split ratio or similarity approach, there is a significant difference in model accuracy between MP = 1 and MP = 5 ($p < 0.001$). However, the difference is not significant for higher values of MP (MP = 30). The post-hoc analysis also showed that the increment in MP has a significant

effect on hit rate and the correct rejection rates ($p < 0.001$). Hit rate increases by 13.85% ($p < 0.001$) from MP = 1 to MP = 5, and increases by 23.7% ($p < 0.001$) from MP = 1 to MP = 30. Similarly, the correct rejection rate increases by 9.22% ($p < 0.001$) for an increase in MP from 1 to 5 and a 13.4%($p < 0.001$) improvement from MP = 1 to Mp = 30. For higher MP values, the improvement in hit rate is much more significant than the improvement in the correct rejection rate. As shown in Figure 3, we can observe a concave relationship between the hit rate and correct rejection rate for MP = 1, whereas this relationship becomes much more linear for MP = 30. There is also a strong interaction effect between similarity approach and MP values. As shown in Figure 2, the models using user perception and perception BERT approaches demonstrate an increase in accuracy for higher values of MP, but MP does not appear to impact the accuracy of models using LSA, GloVe, and canonical BERT models.

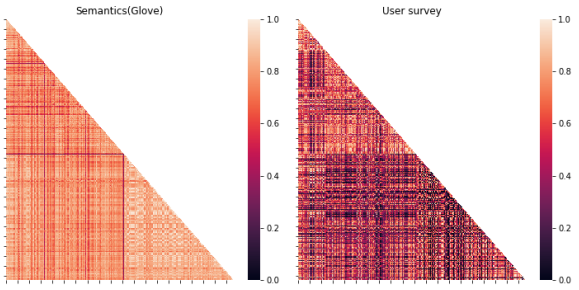

Figure 4: Visualizing similarity between every pair of emails. (Left) Similarity measured using GloVe. (Right) Similarity measured based on user perception

Split Ratio

Finally, we tested whether the IBL model performance depended on the amount of data used for training the agents. We tested two commonly used split ratios: 50-50 and 80-20. Although there was a statistically significant effect of the split ratio on the accuracy and correct rejection rate, as shown in Tables 3 and 4, the improvement was only marginal. On average, model accuracy increased by 1.4%($p < 0.001$) from 50:50 to 80:20 split. The correct rejection rate increased by 1.2%($p = 0.037$). The split ratio did not have a significant effect on the hit rate. There was also no interaction effect between split ratio and similarity approach or between split ratio and MP.

Discussion

Our work shows that the approach used to represent email text within IBL models can significantly affect agent performance in predicting human response. Furthermore, we found that using representations that could consider how humans perceive email messages can have a substantial impact on model performance. We predicted participants' responses to email messages in a laboratory study with 79.7% accuracy by training IBL agents with emails represented using attributes self-reported by participants in the study. In contrast, we achieved 71.7% accuracy by fine-tuning a BERT model to learn similarities between emails based on user-reported attributes. IBL agents using traditional NLP methods (LSA and GloVe) which are effective at capturing semantic similarities, performed better than chance at predicting human response, similar to the results obtained by Cranford and colleagues [9]. Overall, our results show that IBL models are able to adequately predict human response to phishing emails, providing evidence to our hypothesis that people make decisions on phishing messages based on past experiences by activating pertinent memories of decisions made in response to similar emails in the past.

Although methods like GloVe and BERT effectively highlight semantically important features in a piece of text, they may not be effective at highlighting features relevant to people in the email management context. Therefore, as shown in Figure 4, traditional NLP methods like GloVe may end up representing two emails as semantically similar to each other, whereas a human may think otherwise. For example, two emails may contain words with similar co-occurrence frequencies and may include similar latent topics. Yet, they may appear different to a human because they may be communicating two divergent messages. Therefore, more work is necessary to understand how people encode email messages, what salient features of emails are encoded in the memory, and how the features encoded may vary by the type of email (ham vs. mass phishing vs. spear phishing). Understanding these issues could provide us important insights into how humans learn and make decisions on malicious deceptive signals such as phishing emails.

One important implication of this work is that IBL models, or cognitive models in general, could be potentially used in the future to develop personalized phishing probing and training solutions. For example, IBL models could be deployed in email management software such as Microsoft Outlook to actively learn phishing instances that an individual may be susceptible to and may benefit from receiving additional embedded phishing experiences. Furthermore, instead of probing every employee in an organization with generic phishing templates to learn who may be vulnerable to phishing, such human-centered, deep learning-based, and cognitive architecture inspired methods could be effective at quickly detecting human vulnerabilities

within an organization.

This work, however, is not without limitations. The models were developed and validated using a small dataset containing human responses to emails (legitimate and phishing) collected from a laboratory experiment. The dataset may not truly reflect how people would respond to phishing emails in reality. Furthermore, there is a risk of overfitting from fine-tuning BERT using relatively small datasets. Finally, all the models in this work were built using cosine distance function, which may not represent how human partial match two given instances. Therefore, as part of our future work, we intend to test these models on large real-world email datasets and test its effectiveness in adaptive training and risk assessment.

References

[1] Z. M. Hakim, N. C. Ebner, D. S. Oliveira, S. J. Getz, B. E. Levin, T. Lin, K. Lloyd, V. T. Lai, M. D. Grilli, and R. C. Wilson, "The phishing email suspicion test (pest) a lab-based task for evaluating the cognitive mechanisms of phishing detection," *Behavior Research Methods*, pp. 1–11, 2020.

[2] K. Singh, P. Aggarwal, P. Rajivan, and C. Gonzalez, "What makes phishing emails hard for humans to detect?," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 64, pp. 431–435, SAGE Publications Sage CA: Los Angeles, CA, 2020.

[3] A. Aleroud and L. Zhou, "Phishing environments, techniques, and countermeasures: A survey," *Computers & Security*, vol. 68, pp. 160–196, 2017.

[4] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 581–590, ACM, 2006.

[5] M. Wu, R. C. Miller, and S. L. Garfinkel, "Do security toolbars actually prevent phishing attacks?," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 601–610, 2006.

[6] P. Kumaraguru, Y. Rhee, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge, "Protecting people from phishing: the design and evaluation of an embedded training email system," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 905–914, 2007.

[7] P. Kumaraguru, J. Cranshaw, A. Acquisti, L. Cranor, J. Hong, M. A. Blair, and T. Pham, "School of phish: a real-world evaluation of anti-phishing training," in *Proceedings of the 5th Symposium on Usable Privacy and Security*, pp. 1–12, 2009.

[8] B. D. Sawyer and P. A. Hancock, "Hacking the human: the prevalence paradox in cybersecurity," *Human factors*, vol. 60, no. 5, pp. 597–609, 2018.

[9] E. A. Cranford, C. Lebiere, P. Rajivan, P. Aggarwal, and C. Gonzalez, "Modeling cognitive dynamics in (end)-user response to phishing emails," *Proceedings of the 17th ICCM*, 2019.

[10] K. Singh, P. Aggarwal, P. Rajivan, and C. Gonzalez, "Training to detect phishing emails: Effects of the frequency of experienced phishing emails," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 63, pp. 453–457, SAGE Publications Sage CA: Los Angeles, CA, 2019.

[11] C. Gonzalez, J. F. Lerch, and C. Lebiere, "Instance-based learning in dynamic decision making," *Cognitive Science*, vol. 27, no. 4, pp. 591–635, 2003.

[12] D. Morrison and C. Gonzalez, "Pyibl python implementation of ibl."

[13] T. Xu, K. Singh, and P. Rajivan, "Spearsim: Design and evaluation of synthetic task environment for studies on spear phishing attacks," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 65, pp. 1500–1504, SAGE Publications Sage CA: Los Angeles, CA, 2021.

[14] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *European Conference on Machine Learning*, pp. 217–226, Springer, 2004.

[15] P. Rajivan and C. Gonzalez, "Creative persuasion: A study on adversarial behaviors and strategies in phishing attacks," *Frontiers in psychology*, vol. 9, p. 135, 2018.

[16] T. Lejarraga, V. Dutt, and C. Gonzalez, "Instance-based learning: A general model of repeated binary choice," *Journal of Behavioral Decision Making*, vol. 25, no. 2, pp. 143–153, 2012.

[17] P. Aggarwal, F. Moisan, C. Gonzalez, and V. Dutt, "Learning about the effects of alert uncertainty in attack and defend decisions via cognitive modeling," *Human Factors*, p. 0018720820945425, 2020.

[18] J. Anderson and C. Lebiere, "The atomic components of thought lawrence erlbaum," *Mathway, NJ*, 1998.

[19] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, "An integrated theory of the mind.," *Psychological review*, vol. 111, no. 4, p. 1036, 2004.

[20] P. W. Foltz, "Latent semantic analysis for text-based research," *Behavior Research Methods, Instruments, & Computers*, vol. 28, no. 2, pp. 197–202, 1996.

[21] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[24] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 11 2019.

[25] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," *arXiv preprint arXiv:1508.05326*, 2015.

[26] A. M. Law, W. D. Kelton, and W. D. Kelton, *Simulation modeling and analysis*, vol. 3. McGraw-Hill New York, 2000.